

## PL 06: SQL DDL

Para cada um dos problemas, crie, a partir do modelo relacional, o respetivo esquema na base de dados, usando PostgreSQL.

### Problema 1

```

professor(nome)
sala(nome)
turma(nome, nomeSala)
  FK(nomeSala) -> sala
  NN(nomeSala)
leciona(nomeTurma, nomeProfessor)
  FK(nomeProfessor) -> professor
  FK(nomeTurma) -> turma

```

Outras Restrições de integridade:

- RI1. As turmas têm pelo menos um professor
- RI2. Os professores estão associados a pelo menos uma Turma

### Problema 2

```

departamento(nome)
empregado(numero_func, nome, salario, departamento, chefiado_por)
  FK(departamento) -> departamento
  NN(departamento)
  FK(chefeido_por) -> empregado

```

RI: Os diretores não têm nenhum diretor associado e todos os empregados que não são diretores têm, obrigatoriamente, um diretor associado.

### Problema 3

```

pessoa(cc, nome)
aluno(cc, numero_aluno)
  FK(cc) -> pessoa
  UK(numero_aluno)
  NN(numero_aluno)
professor(cc, salario)
  FK(cc) -> pessoa
professor_num_telefone(cc, num_telefone)
  FK(cc) -> professor

```

- RI1. As pessoas têm que ser alunos ou professores, i.e., não podem haver tuplos na relação pessoa que não estejam em professor ou em aluno (especialização total)
- RI2. Uma pessoa não pode ser, simultaneamente, aluno e professor (disjunto)

**Problema 4**

```

comboio_local(numero, tipo_maquina)
comboio_expresso(numero, tipo_maquina)
estacao(nome, localidade)
estacao_expresso(nome)
    FK(nome) -> estacao
estacao_local(nome)
    FK(nome) -> estacao
paragem_exp(numero, nome, hora)
    FK(numero) -> comboio_expresso
    FK(nome) -> estacao_expresso
paragem_loc(numero, nome, hora)
    FK(numero) -> comboio_local
    FK(nome) -> estacao

```

- RI1. Um comboio não pode ser, ao mesmo tempo, local e expresso
- RI2. Uma estação não pode ser, ao mesmo tempo, local e expresso
- RI3. Uma estação tem de ser local ou expresso

**Problema 5**

```

aluno(num_aluno, nome)
exame(exame_id, descricao, data_inicio, data_fim)
disciplina(cod_disc, nome)
disciplina_execucao(cod_disc, seq_id, ano, semestre)
    FK(cod_disc) -> disciplina
notas_exame(num_aluno, exame_id, cod_disc, seq_id, ano, semestre, nota)
    FK(num_aluno) -> aluno
    FK(exame_id) -> exame
    FK(cod_disc, seq_id, ano, semestre) -> disc_d_exec

```